IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF VIRGINIA
RICHMOND DIVISION

THE TRUSTEES OF COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK, *Plaintiff* vs. SYMANTEC CORPORATION, *Defendant*	Civil Action No. 3:13-cv-00808-JRS **JURY TRIAL DEMANDED**

COLUMBIA UNIVERSITY'S RESPONSIVE CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

Page

Page

TABLE OF AUTHORITIES

Page(s)

Cases

I. INTRODUCTION

At the beginning of the meet and confer process, Columbia initially proposed for construction the "emulator" term in the '115/'322 patents because of the particular meaning that is ascribed to that term in the specification. The additional constructions Symantec has proposed are not necessary because a person of ordinary skill in the art can understand and apply the terms at issue. Symantec's constructions do not assist the jury. They simply redraft the claims.

This brief references the Second Declaration of Professor Douglas Szajda ("Szajda Decl. II") and the Declaration of Professor Douglas Szajda ("Szajda Decl. I"). Dkt. 106–1. Exhibits A–W are attached to the Declaration of Gavin Snyder filed with Columbia's opening brief. Dkt. 106–2. Exhibits X–II are attached to the Second Declaration of Gavin Snyder.

II. '544/'907 PATENTS

The parties' main dispute with respect to the '544/'907 patents concerns the meaning of the term "byte sequence feature" and Symantec's related indefiniteness argument. Symantec's positions on both issues are based on the same incorrect interpretation of the specification. Columbia will first address the indefiniteness argument. The discussion of why the claims are not indefinite informs the proper construction of "byte sequence feature."

A. "wherein extracting said byte sequence features from said executable attachment comprises creat[ing/e] a bye string representative of resources referenced by said executable attachment"

Symantec maintains that the '544/'907 patents disclose three ***mutually exclusive*** inventions. According to Symantec "[t]he '544 and '907 patents describe three different types of features . . . namely: (1) 'byte sequences;' (2) 'resource information;' and (3) 'encoded string' features.'" Symantec Brief at 3. This mistaken interpretation of the specification is the premise for Symantec's argument that claims that include resource information as a type of byte sequence feature are indefinite. Symantec is wrong. There is no indefiniteness. Nowhere does the

specification say that a byte sequence feature cannot include information representative of resources referenced by the executable. Indeed, the specification explains the opposite.[1]

There are a large number of claimed inventions in the '544/'907 patents. But one element all of the inventions have in common is the extraction of a byte sequence feature from the executable: the creation of a dossier on the executable. Because a computer program contains various parts, the patent describes how it is possible to build a system that concentrates on a particular part. But these various embodiments are simply examples of information that can be included within byte sequence features.

One strategy for supplying information that can be included in a byte sequence feature is the conversion of the entire executable program into hexadecimal format using a program known as hexdump. This has the benefit of capturing all the data in the executable, including all machine code instructions, for use as features. This is an example of a byte sequence feature.

Another strategy is ***not*** to capture all the data in the program in hexadecimal format, but instead to capture byte sequences with selected information about the executable. For example, a security system can focus on generating "byte strings representative of resources" for use as byte sequence features. ***The specification is definitive that byte strings representative of resources are examples of byte sequence features***: "[a]ccording to another embodiment, extracting the byte sequence features from the executable attachment may comprise creating a byte string

[1] Symantec's indefiniteness arguments turns on a factual question regarding what is disclosed in the specification. To the extent the Court believes Symantec's position presents even a colorable argument of indefiniteness, this is an issue of fact for a jury to decide. *BJ Servs. Co. v. Halliburton Energy Servs., Inc.*, 338 F.3d 1368, 1372 (Fed. Cir. 2003) (indefiniteness, like enablement, has subsidiary questions of fact); *Dow Chem. Co. v. NOVA Chems. Corp. (Canada)*, 629 F. Supp. 2d 397, 404 (D. Del. 2009) ("[T]here may arise cases where genuine issues of fact simply preclude such a treatment. In these circumstances, the question of indefiniteness must be addressed by the trier of fact."). Professor Szajda's declaration makes clear that the facts do not support Symantec's position. Szajda Decl. I, ¶¶ 32–44; Szajda Decl. II, ¶¶ 4–15.

representative of resources referenced by said executable attachment." Ex. C, '544 patent at 3:37–40. The collection of hexadecimal digits generated by hexdump and byte strings representative of resources are ***both*** examples of information that can be included in byte sequence features.

1. The Summary Makes Clear that a "Byte String Representative of Resources" Is an Example of a Byte Sequence Feature

The Summary makes clear that the extraction of byte sequence features is common to the inventions as a whole:

> A byte sequence feature is subsequently extracted from the executable attachment. The executable attachment is classified by comparing said byte sequence feature of the executable attachment with a classification rule set derived from byte sequence features . . . . *Id.* at 3:23–29.

Just as importantly, the Summary of the Invention expressly states that a string of bytes representative of resources is not a separate invention: it is an example of a byte sequence feature:

> Extracting the byte sequence feature from the executable attachment may comprise converting the executable attachment from binary format to hexadecimal format. According to another embodiment, extracting the byte sequence features from the executable attachment may comprise creating a byte string[2] representative of resources referenced by said executable attachment. *Id.* at 3:34–40.

Symantec's indefiniteness argument cannot be correct in light of this express teaching of an embodiment in which byte sequence features include byte strings representative of resources.

2. The Detailed Description Makes Clear that a "Byte String Representative of Resources" Is an Example of Information within a Byte Sequence Feature

The Detailed Description of Exemplary Embodiments section of the patents confirms that a "byte string representative of resources" is a specific type of "byte sequence feature," not an

[2] The specification interchangeably refers to byte sequences as strings of bytes, including in the hexdump process. *See, e.g.*, Ex. C, '544 patent at 13:22–25; 15:21–29; 16:34–37.

alternative embodiment.

The Detailed Description walks the reader through an example of an inventive process disclosed by the patent. *Id.* at 5:16–17. It notes that "[t]he next step of the method is to extract features from each executable (Step 20)." *Id.* at 5:57–58. It then describes byte sequence features:

> The next step of the method is to extract features from each executable (Step 20). Features . . . are defined as properties extracted from each example program in the data set, e.g., byte sequences, that a classifier uses to generate detection models. (Signatures, as distinguished from features, typically refer to a specific feature value, while a feature is a property or attribute of data (such as "byte sequence feature") which may take on a set of values. *Id.* at 5:57–64.

Step 20 in Figure 1 is entitled "Extract Features from Data." *Id.*, Fig.1. This corresponds to the claim limitation "extracting a byte sequence feature." It is the creation of the dossier on the executable. The specification then gives different ways of implementing byte sequence extraction step 20.

One way is using the hexdump program. *Id.* at 6:7–22. The specification then describes additional methods "to carry out step 20." *Id.* at 6:23–24. One alternative is to focus only on a particular subset of data in the executable: "it is possible to extract a set of features . . . representative of resources referenced by the binary." *Id.* at 6:55–58. The specification is explicit that the extraction of byte strings representative of resources relates to the creation of byte sequence features, introducing its discussion with the qualifier "[a]ccording to ***another approach to feature extraction***." *Id.* at 6:26–29 (emphasis added). The byte sequence features representative of resources discussed in this section of the specification are ***not*** machine code instructions, as discussed in paragraph 43 of Professor Szajda's first declaration.[3]

[3] Another source of byte sequence features is data generated by a GNU strings program (Ex. C, '544 patent at 7:48–54), discussed by Professor Szajda at paragraphs 16–18 of his second declaration.

Each of these above strategies (and others disclosed in the specification) is a way of supplying information that can be included within a "byte sequence feature."

3. The Claims as Originally Filed Make Clear That a "Byte String Representative of Resources" Is an Example of Information within a Byte Sequence Feature

The claims as originally filed with the application that became the '544/'907 patents further establish that a "byte string representative of resources" is an example of information that can be included in a "byte sequence feature." The claims as originally filed with the parent application are part of the written description of the patent. *Crown Packaging Tech., Inc. v. Ball Metal Beverage Container Corp.*, 635 F.3d 1373, 1380 (Fed. Cir. 2011) ("Original claims are part of the specification and in many cases will satisfy the written description requirement.").

In the claims as originally filed there are three independent claims. Independent claims 1 and 29 recite the element "extract[] a byte sequence feature from said executable attachment." Ex. X, U.S. Patent Application No. 10/208,432, at 11–12. Depending off each of these independent claims is a claim reciting that as part of byte sequence feature extraction "a byte string representative of resources referenced by said executable attachment" is created. *Id.*, claims as filed 4, 32. This establishes that a "byte string representative of resources" is information that can be included in a "byte sequence feature."

4. Claim 28 Does Not Support Symantec's Indefiniteness Position

Symantec argues that claim 28 of the '544 patent supports its incorrect interpretation of the specification that a byte sequence feature cannot include byte strings representative of resources. Symantec Brief at 10. Claim 28 recites "a feature extractor configured to extract a byte sequence feature from said executable attachment, wherein said feature extractor is further configured to create a byte string representative of resources referenced by said executable attachment." Ex. C, '544 patent at claim 28. Symantec believes that this limitation describes

two unrelated operations: first, a feature extractor creates a byte sequence feature (which Symantec interprets as being limited to machine code instructions); second, the same feature extractor does something completely different: it creates a byte string representative of resources.

The structure of the claim demonstrates that Symantec is incorrect. A claim limitation requires a feature extractor that can create features using at least one source of information. The "wherein" clause restricts the feature extractor: it is not enough for the feature extractor to be able to use just any type of information to generate byte sequence feature information. It must be able to generate byte sequence features using resource information. The limitation exists in its current form because claim 28 is a combination of independent claim 29 as filed and dependent claim 32 as filed. Ex. X, at 12. Moreover, the next element of claim 28 is a rule evaluator that uses the byte sequence features in classification. The element does not mention resource information. If Symantec's interpretation were correct, the byte string representative of resources would be created, but then discarded without being used. That interpretation makes no technical sense.

5. Symantec's Dr. Jaeger Contradicts the Specification

Symantec's expert Dr. Jaeger provides very few scientific opinions. Instead, he simply repeats the legal arguments made by Symantec. *See, e.g.*, Jaeger Decl., ¶ 22. He does, however, make an assertion that confirms his and Symantec's incorrect understanding of the patents. Column 13 of the patents states "[i]t is understood that the feature extraction step described herein [hexdump] is alternatively performed with a binary profiling method in another embodiment." Ex. C, '544 patent at 13:29–32. According to Dr. Jaeger, the hexdump embodiment is the only embodiment that creates byte sequence features, and this is distinct from collecting data on resources. Both Symantec and Dr. Jaeger call the latter "binary profiling." Jaeger Decl., ¶ 22; Symantec Brief at 4.

Dr. Jaeger's opinion expressly contradicts the specification. "Binary profiling" is the general name for the process that yields byte sequence features. "Binary profiling" expressly includes both the output of hexdump and the creation of byte strings representative of resources. The specification states that "FIGS. 2–4 illustrate a ***several approaches to binary profiling***.'" Ex. C, '544 patent at 4:20–21 (emphasis added). Figure 2 is an example of hexadecimal data from hexdump which Dr. Jaeger maintains is a byte sequence feature. Compare *id.*, Figure 2 and 6:14–17; Jaeger Decl., ¶ 17. Figures 3–4 are byte sequence features containing resource information. Compare Ex. C, '544 patent Figure 3 and 7:6–11; Figure 4 and 7:18–23.

B. "byte sequence feature"

As discussed above in Section II.A , the patents disclose a number of different types of information that can be included in a "byte sequence feature." The claim term is not limited to machine code instructions as Symantec contends.

One class of information is the output of the hexdump program. Hexdump converts the entire program into hexadecimal strings. Szajda Decl. I, ¶¶ 43–44. In the words of the specification, "each byte sequence in the program is used as a feature." Ex. C, '544 patent at 6:21–22. Because hexdump converts the entire executable, its output includes representations of machine code instructions, as well as everything else in the program. Szajda II Decl., ¶¶ 12–15. But the output of hexdump is only one of multiple sources of information that can be included in byte sequence features. This is made explicit at column 3, lines 3–40 of the '544 patent, which teaches that byte strings representative of resources are byte sequence features.

1. The Plain Meaning of "Byte Sequence Feature" Is Not Limited to "Machine Code Instructions"

Symantec argues that Columbia's construction is not productive because it "provides no insight to the trier of fact concerning the technical term 'byte sequence,' . . . ." Symantec Brief

at 6. But "byte sequence" is not a controversial or overly-technical phrase, and does not require any additional gloss. Szajda Decl. I, ¶¶ 36–38. Dr. Szajda explains exactly how it describes both the hexadecimal features depicted in Figure 2 and the strings representative of resources in Figure 3. Szajda Decl. II, ¶¶ 5–11. The dispute is not about assisting the jury. The dispute is about Symantec's desire to import a machine code instruction limitation into the definition.

2. Symantec Misinterprets the Portion of the Specification that Discusses Machine Code Instructions

In arguing that byte sequence features should be limited to machine code instructions, Symantec improperly seeks to limit the claim terms to one particular embodiment. Indeed, the primary passage Symantec relies on from column 6 begins with the words "[i]n the exemplary embodiment . . . ." Ex. C, '544 patent at 6:7–22. Symantec's argument is wrong for two reasons. First, using embodiments to limit the scope of the specification violates explicit Federal Circuit precedent. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1323 (Fed. Cir. 2005) (en banc) ("[W]e have expressly rejected the contention that if a patent describes only a single embodiment, the claims of the patent must be construed as being limited to that embodiment.").

And second, Symantec misunderstands the hexdump embodiment. The embodiment creates strings of bytes out of all information in an executable. Szajda Decl. I, ¶¶ 43–44; Szajda Decl. II, ¶¶ 12–15. This is not limited to machine code instructions. Symantec cites two portions of the specification that it contends show limitation to machine code instructions, one at column 6 and another at column 13. Symantec Brief at 6–7.

Column 6 states that "[i]n the exemplary embodiment, hexdump was used in the feature extraction step . . . . The byte sequence feature is informative because it represents the machine code in an executable." Ex. C, '544 patent at 6:7–14. Symantec appears to believe that this passage ***defines*** byte sequence features as machine code instructions. It does nothing of the sort.

The passage is instead discussing the fact that in "the exemplary embodiment" using hexdump, byte sequence features may be taken from any part of the executable, which necessarily includes any of the machine code instructions.

Column 13, lines 13–37 states that when hexdump is used to create the byte sequence features and that "[t]his byte sequence feature is useful because it represents the machine code in an executable." *Id.* at 13:24–26. The specification is not stating that a byte sequence is only machine code instructions. To the contrary, the hexdump strategy analyzes all information in an executable, including resource information: "this [hexdump] approach ***involves analyzing the entire binary***, rather than portions such as headers. . . ." *Id.* at 13:26–29 (emphasis added). The hexdump embodiment does not analyze only the machine code instructions in the executable. It analyzes the entire executable.

3. Provisional Applications Cannot Alter the Issued Patents

Manuscripts generated by Professor Stolfo's laboratory were submitted as provisional applications and are referenced on the face of the '544/'907 patents: Provisional Application Nos. 60/308,622 and 60/308,623. The actual specification for the '544/'907 patents was re-written significantly from the text of the draft manuscripts.[4] Symantec references portions of the provisional application to support its incorrect interpretation of the relationship between byte sequence features and byte strings representative of resources. Symantec Brief at 7. The Federal Circuit holds that it is the language and teaching in the final specification that controls for claim construction. *Sun. Pharm. Indus., Ltd. v. Eli Lilly & Co.*, 611 F.3d 1381, 1388 (Fed. Cir. 2010) ("[T]he relevant specification for claim construction purposes is that of the issued patent, not an early version of the specification that may have been substantially altered . . . .").

[4] The fact that the provisional was so substantially re-written is relevant to claim construction but is not relevant to the question of priority date.

In any case, the provisional applications actually support Columbia's position. The applications make clear that the purpose of the hexdump embodiments is to capture information on the entire program, not just machine code instructions. Ex. 2, U.S. Provisional Application No. 60/308,622, at 5 ("[A]nalyzing ***the entire binary*** gives more information for non-PE format executables than the strings method.") (emphasis added).

C. "email interface"

The specification teaches that the email interface in the '544/'907 patents can perform a large number of potential functions. See Szajda Decl. I, ¶ 46; Ex. C, '544 patent at 15:30–37 and claims 32, 41, and 42. Symantec has arbitrarily limited email interface to only one of those functions—reintegration. This makes no sense. Many of the functions described above do not require reintegration. *See* Szajda Decl. I, ¶ 46. Indeed, some embodiments are mutually exclusive with email reintegration. For example, one of the identified functions in the specification for the email interface is to "quarantine the email" (*see* Ex. C, '544 patent at 15:35–36), the exact opposite of email reintegration.

Figure 9 of the specification also makes clear that the email interface can perform a number of different functions. Block 232 in the figure shows arrows pointing toward and away from email traffic, and arrows toward and away from the email filter. This is consistent with the fact that the email interface performs many different functions in different embodiments. Symantec simply picks one function that involves the interface reintegrating email into email traffic and arbitrarily uses that to define the interface.

Symantec complains that Columbia's construction is deficient because it "does not describe what those components [that the email interacts with] are or do." Symantec Brief at 11. But claims 32, 41 and 42 of the '544 patent, the only claims where email interface appears, describe exactly what function the email interface performs.

III. '084/'306 PATENTS

A. "probabilistic model of normal computer system usage" / "normal computer system usage"

Symantec's proposed redrafting of "normal" to mean "typical, attack free" is not designed to add meaning to a hard to understand concept. Nor is it designed to clarify what a model is, or what computer system usage is. Instead, the redrafting is designed as a hook for the insertion of a negative limitation that appears nowhere in the claims: that when constructing a model of normal computer usage (or for that matter, "typical, attack free" computer system usage) based on examples of normal accesses to the registry, the security system can never supplement its use of ***normal*** accesses with information on ***abnormal*** accesses to the registry.

This negative limitation finds no support in the claims or the specification. It is also directly contrary to the materials incorporated into the specification and the prosecution history, which expressly contemplate the use of data sets that involve normal accesses ***supplemented with abnormal accesses*** to construct a model of normal behavior.

Models of normal behavior are specified by the claims, and those models must use data on normal behavior. But nothing in the claims or specification limits the use of additional types of data in constructing the model of normal behavior. As discussed in the Szajda Decl. I, paragraphs 57–72, it is accepted in the field that information on normal computer system accesses supplemented with abnormal system accesses can be used to construct a ***model of normal behavior***.

1. The Claims of the '084/'306 Patents Do Not Exclude Supplementary Use of Abnormal Accesses

The '084/'306 patent claims make clear what must be present in systems that practice the disclosed inventions. For example, the '084 patent claim 14 requires that the system must employ a "model of normal computer system usage based on records of a plurality of processes

that access the operating system registry and that are indicative of normal computer system usage." Ex. E, '084 patent at 23:36–39.

There is no dispute that these elements must be present. The dispute turns on Symantec's desire to add an additional limitation to the claim "***and the model on normal behavior cannot use any supplemental abnormal accesses***." This limitation, of course appears nowhere in the claims. Indeed, although the claims do include a discussion of data the model must include—"records of a plurality of processes that access the operating system registry and that are indicate of normal computer system usage"—Symantec is ***not*** asking the Court to construe this phrase to exclude the use of other types of data. Instead, Symantec appears to contend that the specification somehow disclaims any system that supplements the data set used to construct its model of normal behavior so that in addition to including normal activity information, it also includes abnormal activity information. Symantec Brief at 14–17.

2. There Is No Clear and Unmistakable Disclaimer of Methods that Supplement Normal Accesses with Abnormal Accesses

Symantec's proposed construction should be rejected as an initial matter because it is inconsistent with governing law. The Federal Circuit is clear that in order to import a negative limitation into a claim such as Symantec proposes there must be a "clear and unmistakable" disclaimer. *Teleflex, Inc.v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1325 (Fed. Cir. 2002); *Thorner v. Sony Computer Entm't Am. LLC*, 669 F.3d 1362, 1366 (Fed. Cir. 2012) ("[E]ven a direct criticism of a particular technique [does] not rise to the level of a clear disavowal . . . . [E]ven where a particular structure makes it 'particularly difficult' to obtain certain benefits of the claimed invention, this does not rise to the level of disavowal of the structure.").

In *Thorner*, "the specification repeatedly use[d] the term 'attached' in reference to embodiments where the actuators are 'attached to [an] *outer* side' … and "never use[d] the word

'attached' when referring to an actuator located on the *interior* of a controller." *Thorner*, 669 F.3d at 1367. And when the specification discussed connection to an inner surface, they consistently used the term "embedded within." *Id.* The Federal Circuit held that the consistent distinction between attachment and embedded did not "rise to the level of . . . disavowal." *Id.* at 1367. The court refused to limit the claim term "attached" to mean "attached to an outer surface" and adopted the term's broader plain meaning—"attached to either the interior or exterior." *Id.* at 1368.

Here, the Abstract and Summary of the '084/'306 patents contain no statement clearly and unmistakably disclaiming the supplementation of normal activity information with abnormal activity information.

The Detailed Description of the Exemplary Embodiments in the patents also contains no statement clearly and unmistakably disclaiming the supplementation of normal activity information with abnormal activity information. Symantec's brief focuses on the fact that in one of the embodiments in the Detailed Description, the inventors noted "[i]f the model of the normal registry behavior is trained over clean data, then these kind of registry accesses [system installation] will not appear in the model." Ex. E, '084 patent at 6:30–33. This statement, however, is not a disclaimer. The language, which appears in the "Exemplary Embodiments" section, is merely discussing an embodiment. The sentence also describes a conditional circumstance: "if a model is trained . . . over clean data." *Id.* This does not mean that all models have to be trained to assess system installation activity. And the inventors openly stated "that various modifications can be made by those skilled in the art without departing from the scope and spirit of the invention." *Id.* at 17:55–58. *See Golight, Inc. v. Wal-Mart Stores, Inc.*, 355 F.3d 1327, 1331 (Fed. Cir. 2004) ("[P]atentees [are] not required to include within each of

their claims all of [the] advantages or features described as significant or important in the written description.").

Moreover, even if one assumes for the sake of argument that a data set with no abnormal access information is the only embodiment described (it is not), the Federal Circuit consistently holds that the features of even the sole embodiment in a specification cannot limit the claims of a patent. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1323 (Fed. Cir. 2005) (en banc) ("[W]e have expressly rejected the contention that if a patent describes only a single embodiment, the claims of the patent must be construed as being limited to that embodiment.").

3. The Inventions in the '084/'306 Patent Are Not Directed to Using Only Normal Accesses

Symantec's fixation on importing a negative limitation into the claims of the patent forbidding any consideration of supplemental abnormal accesses is perplexing. The basic concepts of a model of normal behavior and using data on normal activity in this model were not invented by the '084/'306 patents. They are in the prior art. The patents address more specific issues, for example using probabilistic models to accurately predict whether a very unique class of computer functions—e.g., operating systems registry accesses—are anomalous. Whether supplemental abnormal registry accesses are or are not used as part of model construction using normal registry accesses has no relevance to the inventive focus of the patents.

The context in which the inventors performed their research is discussed in the Background of the Invention. The standard prior art strategy used databases that contained ***exclusively*** information on malicious programs: "these algorithms match host activity to a database of signatures which correspond to known attacks. This approach . . . ***requires previous knowledge of an attack*** and is rarely effective." Ex. E, '084 patent at 2:28–32.

The inventors noted that a significant improvement over this standard technique was the

use of anomaly detection that constructed a model of normal computer system usage that, unlike the previous methods which used ***exclusively*** malicious data, considered data on normal activity: "[a]nomaly detection algorithms may build models of normal behavior in order to detect behavior that deviates from normal behavior and which may correspond to an attack." *Id.* at 2:34–37. The inventors then provided references that describe how models of normal behavior can be constructed. *Id.* at 2:39–64. These references include a number of publications by the inventors themselves. As discussed in Section III.A.4 below and paragraphs 67–72 of Prof. Szajda's first declaration, these references specifically teach the construction of models of ***normal*** behavior using data on ***normal*** activity, but ***supplemented*** with data on ***abnormal*** activity. In other words, the specification discloses models of normal computer system usage constructed using both normal and abnormal accesses.

The inventors explained that their goal is to improve upon pre-existing models of normal behavior. *Id.* at 2:65–3:7. Two areas of improvement are relevant to claim construction. The first relates to the observation that the activity in an operating system registry (a unique data structure in Windows) can serve as an extraordinarily sensitive tool for detecting anomalous activity.[5] *Id.* at 3:31–42; 5:60–6:9; 6:16–24. The second relates to the use of a "probabilistic model" that can determine whether a newly-observed process is malicious. *Id.* at 3:43–54. Neither of these areas have anything to do with whether information on normal activity used to construct the model of normal behavior is supplemented with information on abnormal activity.

In the Detailed Description section of the patents the inventors describe the experiments that allowed them to make the critical discoveries discussed in the preceding paragraph. *Id.* at 15:44–17:43. But as the Federal Circuit has held repeatedly, inventions disclosed in a patent are

[5] In an alternative embodiment, the specification focuses on particular information in the file system. Ex. E, '084 patent at 17:43–55.

not limited to the particular systems used to perform the experiments presented in the patent. *Golight*, 355 F.3d at 1331 ("Moreover, '[a]bsent a clear disclaimer of particular subject matter, the fact that the inventor anticipated that the invention may be used in a particular manner does not limit the scope to that narrow context.'") (internal citations omitted).

Large sections of Symantec's brief are devoted to limiting the claims to embodiments constructed by the inventors when they performed their experiments. Indeed, this is the focus of Symantec's arguments regarding the provisional application listed on the face of the '084/'306 patents: 60/351,857. Symantec Brief at 15. The provisional application is not a formal patent application per se but a copy of a manuscript.[6] But even in the early manuscript it is clear that the inventors' substantive conclusions had nothing to do with whether ***only*** normal accesses were used to construct the model of normal activity. The conclusion of the article confirms that the focus is on establishing the importance of considering registry data: "By using registry activity on a Windows system we were able to label all processes as either attacks or normal." Ex. 6, at COL00007585. The article does not teach that supplemental abnormal accesses must always be excluded. The article teaches the importance of registry data, regardless of whether it is from normal accesses or abnormal accesses: "Most importantly we have shown that a system that uses only registry data can be effective as an intrusion detection system." *Id.* The inventors' experiments proved a very important principle: that probabilistic models based on registry activity are powerful tools for detecting anomalies. These are some of the concepts reflected in the claims, not the arbitrary exclusion of supplemental abnormal accesses.

Symantec fixates on the fact that in certain experiments in the article the inventors allegedly used 100% clean data—i.e., no supplemental data on abnormal activity. But the

[6] The fact that the provisional was so substantially re-written is relevant to claim construction but is not relevant to the question of priority date.

inventors were absolutely clear that the focus should not be on the actual implementation described. The inventors chose the allegedly 100% clean data embodiment purely for expediency: "We want to emphasize that a more sophisticated algorithm can be used in place of this heuristic approach which was chosen because it is simple and efficient." *Id.* at 10.

4. Inventor Eskin's Article Describes a Model of Normal Behavior Constructed from Normal Accesses, but Supplemented with Abnormal Accesses

After introducing the concept of an anomaly detection system that employs "models of normal behavior," the specification then provides examples of models that can be used as the foundations for the critical improvements described in the patents. Ex. E, '084 patent at 2:34–37; 2:38–64. These examples include published work by a number of the inventors that describe the construction of models of normal behavior based on normal activity data but supplemented with data on abnormal activity.[7] Szajda Decl. I, ¶¶ 67–72.

Although Symantec admits that one of these papers, by Inventor Eskin, describes the use of data on normal activity supplemented with abnormal activity, it claims that the paper is not describing the construction of a model of normal behavior. Symantec Brief at 17. Symantec is incorrect. The paper states definitively that it is constructing a model of normal behavior: "[t]he anomaly detection method presented in this paper makes three important assumptions. The first assumption is that ***normal data can effectively be modeled using the probability distribution***." Ex. N, at 6 (emphasis added). The paper is also definitive that it based this model on normal accesses, but ***supplemented*** with other information: "[t]he data provides normal and ***intrusion***

[7] *See, e.g.*, Ex. P, at COL00009367 ("First, sequences of *n* consecutive system calls were . . . supplied to a machine learning algorithm to learn the patterns of 'normal' and 'abnormal' sequences. These patterns can then be used to examine a new trace and determine whether it contains sufficient abnormal sequences to be identified as an intrusion (anomaly).").

traces of system calls for several processes." *Id.* at 4 (emphasis added).

The article explains that it is not necessary to ensure that the data used to construct a model of normal activity includes only 100% normal accesses. Using data made up of 100% normal access is expensive. And in the designs presented, using a data set that includes normal accesses supplemented with abnormal accesses achieves better results: "since anomalies are extremely rare, … we can use the much simpler direct approach presented in this paper." *Id.* at 1. The article emphasizes the benefit of supplementing with abnormal activity in certain embodiments: "the anomalies themselves can be of interest as they may show rarely occurring events." *Id.* And the article also teaches that the use of normal activity supplemented with abnormal activity provides benefits when used in a probabilistic model. *Id.* at 4.

Symantec contends that this paper "defines" "normal data" as 100% clean. Symantec Brief at 17. Although it is not reasonable to assert that an article written in 2000 was intended to use the same terminology in the same way as a patent written three years later, even accepting the premise of Symantec's argument, there is still no disclaimer. The patent claims do not use the term "normal data" when describing ingredients used by the model. The claims refer to, for example, "access. . . indicative of normal . . . usage." Ex. E, '084 patent, claim 14. The Eskin article consistently describes building a model of normal usage by using data on normal accesses as part of a data set that includes supplemental abnormal data. *See* Ex. II The inventors expressly directed the readers of their patents to methods of constructing models of normal activity that involve the use of normal accesses supplemented with abnormal accesses.

5. Application 10/352,342 Confirms that Data Sets Can Supplement Normal Data with Abnormal Data

The specification incorporates by reference Application 10/352,342 for the purpose of giving examples of data sets that can be used to construct the models of normal behavior used in

the Detailed Description section of the specification. Ex. E, '084 patent at 14:3–10 ("[M]odel[s] that will represent normal usage" can use "[t]he database [that] is described in in greater detail i[n] concurrently filed U.S. application Ser. No. 10/352,342 . . . which is incorporated by reference in its entirety herein."). Symantec represents that the '342 application is limited to designs in which no information on abnormal activity is used to supplement information on normal activity. Symantec Brief at 15. This representation is incorrect. The '342 application makes clear that information on normal activity supplemented with information on abnormal activity can be used as data sets.

The '342 application notes that, in the prior art, "most" (not all) anomaly detection systems trained on "purely normal" data. Ex. 8, '342 application at ¶ 0010. It also notes that the use of a purely normal data set can be undesirable: "[t]his data can be very expensive because the process of manually cleaning the data is quite timing consuming. Also, some algorithms require a very large amount of normal data which increases the cost." *Id.*

The '342 application notes that its designs allow for the use of "heterogeneous data." *Id.* at ¶ 0029. The specification then describes a "data warehouse" that contains a "data [set] of interest … which is suspected to contain intrusions." *Id.* at ¶ 0078. The disclosure of data sets that include normal accesses supplemented with abnormal accesses is significant because the '084/'306 patents reference the '342 application with regard to "database[s]" used to construct the model. Ex. E, '084 patent at 14:5–10.

Symantec quotes a passage from the '342 application in its brief that describes one form of anomaly detection that involves "data that contains no intrusions." Symantec Brief at 15. Symantec's focus on this passage is notable for two reasons. First, the inventors clearly knew how to write the phrase "data that contains no intrusions." That phrase does not appear in the

'084/'306 patents claims. Second, Symantec ignores the discussion in the '342 application that appears two paragraphs later, which describes "anomaly detection algorithms" which use data sets that include normal data supplemented with abnormal data: "intrusions are very rare compared to the normal data and they are also quantitatively different. Because of this, intrusions are outliers in the data and can be detected." Ex. 8, '342 application at ¶ 00106.

The '342 application confirms that its diverse data sets "can support a variety of different intrusion detection systems. One example . . . is the Registry Anomaly Detection (RAD) system, which is described in greater detail in [the application that would become the '084/'306 patents, which was filed concurrently with the '342 application]." *Id.* at ¶ 00123.[8]

The simple fact is that the '342 application, which the '084/'306 patents incorporate by reference to provide examples of data sets that can be used to create models of normal behavior, expressly describes the use of normal accesses supplemented by abnormal accesses.[9]

6. The Prosecution History Also Confirms that the Claims Encompass Normal Data Supplemented with Abnormal Data

Symantec represents that, during the '084 patent prosecution, Columbia asserted that the Chong reference used both normal and abnormal activity to construct its model, and that this was not covered by the claims. Symantec claims that the PTO allowed the claims on this basis. Symantec Brief at 15–16. These representations are not accurate. The PTO made clear that the

[8] See Ex. Y, U.S. Patent No. 7,225,343, 24:49–59 (patent that issued based on '342 application which references the '084/'306 patent specification in the above quoted paragraph). Compare Ex. E, '084 patent at 4:55–64 (noting that the exemplary embodiment discloses RAD).

[9] In advance of claim construction, Symantec sought to take the deposition of one of the named inventors, Andrew Honig. Mr. Honig works for Google and had no substantive interactions with anyone affiliated with Columbia before his deposition. Ex. Z, Honig Dep., 10:13–11:9. He was definitive that the incorporated 10/352,342 application, as well as the publications of his co-inventors referenced in the '084/'306 patent, make clear that models of normal behavior can be constructed with normal data supplemented with abnormal data. *Id.* at 202:21–203:3; 203:12–21; 204:13–205:8; 205:14–23; 206:24–207:2; 207:10–15; 207:22–25; 208:17–22; 208:23–209:11; 194:13–195:9; 196:7–197:5.

claims encompass systems that use normal data supplemented with abnormal data to build models of normal behavior.

The Chong reference only discloses the use of ***abnormal*** information. It describes collecting data on "attacker type," "attack objective," "attack intent," "attacker location," "attack methods," "target type," and "probing activity." Ex. 29, US2003/0070003 ("Chong") at ¶ 0021; *id*. at ¶ 0029 ("[T]he models can be used to characterize, for example, the nature, type, and objectives of a computer network attack based on the observable evidence by generating attack assessment hypothesis and probabilities."). Columbia therefore argued during prosecution that "Chong provides no teaching relating to whether processes are 'normal' . . . ." Ex. AA at COL0000957. In other words, according to Columbia, the deficiency of Chong was not that it considered *some* abnormal information. The deficiency was that it considered *no* normal information. Columbia also presented a number of alternative arguments. *Id.* at COL0000958.

The PTO initially rejected all of these arguments. Of particular relevance, the PTO noted that Chong did not limit in any way the type of information that could be considered: "Chong states in paragraph 0014 that any information corresponding to events associated with the computer network may be collected." *Id.* at COL0000966. If the claims of the '084/'306 patents were limited to systems that used ***only*** normal information, the PTO could never have maintained the rejection based on Chong. ***The PTO did not state that Chong disclosed the use of exclusively normal information. The PTO believed that Chong only disclosed the generic use of normal and abnormal information combined. This was sufficient because the claims of the '084/'306 patents do not require the use of only normal information in the creation of models***.

Consistent with the PTO's position that a reference which allegedly disclosed the use of

both abnormal and normal information together is within the scope of the claims, Columbia instead argued in its next response that Chong did not anticipate because it "neither discloses nor suggests a technique for determining the likelihood of observing an event which was not observed during the gathering of features" from the operating system registry. *Id.* at COL0000990. The PTO agreed.

In its next Office Action, the PTO allowed those claims that included the limitation "gathering features from records of normal processes *that access the operating system registry* and determining the likelihood of observing an event that was not observed during the gathering of features from the records of normal processes." *Id.* at COL00001236–1237 (emphasis in original). The PTO refused to allow claims that did not include the underlined language.

Prosecution history disclaimer only exists when there is a "clear and unmistakable" disclaimer. *Golden Bridge Tech., Inc. v. Apple Inc.*, No. 2013-1496, 2014 WL 3397224, at *2 (Fed. Cir. July 14, 2014). Columbia never disclaimed systems that included supplemental abnormal information. To the contrary, the PTO consistently rejected claims during prosecution based on a prior art reference (Chong) that generically disclosed the use of normal and abnormal information together. Columbia overcame the Chong reference on a completely different basis. See *Micro Chem., Inc. v. Great Plains Chem. Co., Inc.*, 194 F.3d 1250, 1260–1261 (Fed. Cir. 1999) (no disclaimer when argued distinction over prior art was not basis for allowance).

The claims, the specification, the articles and applications referenced in the specification, and the prosecution history, establish that there is no clear and unequivocal disclaimer of systems that build models of normal behavior based on normal data but supplement with abnormal data.

B. "anomaly"/"anomalous"

Symantec's insertion of the concept of a "model of normal behavior" (or for that matter a model of "typical, attack-free" behavior) into the definition is inconsistent with the specification

and the accepted meaning of anomaly in the art. Columbia Brief at 23–24. Symantec claims that Columbia's construction is confusing because the test used to assess whether an anomaly has occurred is not specified in the definition. Symantec Brief at 20. But a definition is not the same as the test used to determine infringement.[10] The claims make clear that the inventive systems use models of normal behavior to detect anomalies. There is no need to add redundancy into the claim.

C. "operating system registry"

The operating system registry is a unique structure in Windows. The monitoring of this unique structure is one of the central insights of the patent. The Summary of the Invention states "[i]t is another object of the invention to generate a model of the normal access to the Windows registry. . . ." Ex. E, '084 patent at 3:16–20. Indeed, the specification expressly defines the operating system registry as the Windows registry. *Id.* at 4:55–64 ("Microsoft™ Windows™ registry (hereinafter referred to as the 'Windows™ registry' or the 'registry')").[11] The specification contrasts the unique data structure in Windows known as the operating system registry with file systems that host configuration information. *Id.* at 17:43–54.

There is no artifice to Columbia's construction. The specification makes clear that there

[10] *PPG Indus. v. Guardian Indus. Corp.*, 156 F.3d 1351, 1355 (Fed. Cir. 1998) ("Claims are often drafted using terminology that is not as precise or specific as it might be . . . . That does not mean, however, that a court, under the rubric of claim construction, may give a claim whatever additional precision or specificity is necessary to facilitate a comparison between the claim and the accused product . . . . [T]he task of determining whether the construed claim reads on the accused product is for the finder of fact.").

[11] The claims as filed with the original parent application to the '084/'306 patent actually used the term "Windows™ registry." Ex. AA at COL000180–86. Because trade names are not allowed in patent claims, and because the specification defined "registry" as the "Windows™ registry," Columbia was able to amend the claims to read "operating system registry" without altering claim scope. *See* Ex. AA, at COL00000801. The rule against trade names in claims is meant to prohibit ***broadening*** of claim scope, as when a trade name stays the same but the technical basis for the named product changes. *See Ex parte Simpson*, 218 U.S.P.Q. 1020 (Bd. App. 1982).

is a distinction between the registry and a generic file system that contains configuration data such as in a Linux or Unix system. *Id.* at 17:47–52. The distinction is not the presence of configuration data. Linux and Unix file systems contain configuration data. Szajda Decl. I, ¶ 55. What makes the registry unique is that it is a hierarchical tree structure in which the data is organized as key/value pairs. Columbia's construction tracks these accepted defining elements.

For example, a research article entitled "Forensic analysis of the Windows registry in memory" makes clear two points: First, "[t]he Windows registry is a hierarchical database used in the Windows family of operating systems to store information that is necessary to configure the system." Ex. BB at S26.

Second, what distinguishes the registry structure from a file system is this hierarchical structure with the key/value format: "As for the structure of the registry data itself, it is generally composed of two distinct data types: key nodes and value data." *Id.* at S27. These are the important differences with a file system: "data in the registry always has an explicit associated type, whereas data on a filesystem is generally only weakly typed (for example, through a convention such as file extension)." *Id.*

A summary of the registry by Microsoft also confirms that Columbia's construction captures the defining features of the registry:

> The registry is a hierarchical database that contains data that is critical for the operation of Windows and the applications and services that run on Windows. The data is structured in a tree format. Each node in the tree is called a key. Each key can contain both subkeys and data entries called values. Ex. CC.

Symantec also criticizes Columbia for not including additional details on the information that is stored in keys. Symantec Brief at 21. For example, the specification states that "[t]he registry is the main storage location for . . . ." Ex. E, '084 patent at 5:29–35. But statements about what may be in the keys do not assist the jury in understanding what the registry is.

Columbia's construction reflects the defining features of the registry.

IV. '115/'322 PATENTS

A. "emulator"

The '115/'322 patents specify the use of a software/hardware component that permits the monitoring and selective execution of all or part of a program. Columbia Brief at 27–30. In the disclosure of the patents, this structure is called an emulator. Using an emulator to selectively execute code can avoid the potentially inefficient step of analyzing an entire program in an artificial environment while attempting to detect anomalies. Ex. G, '115 patent at 13:14–24; 9:12–15; Fig. 6, block 610; 16:47–17:24. By monitoring code execution, an emulator can also prevent malicious code from damaging a system. *Id.* at 10:8–20; 10:27–41. Columbia's construction shows fidelity to the essential features of an emulator that must be present to realize the inventions described in the patents. Other than these essential features of monitoring and selective execution, the patents place no limits on the emulator. For example, the inventors working embodiment employed a "Selective Transactional Emulation . . . using the Valgrind emulator" but they were equally clear that "using any other suitable technique" was acceptable to achieve the goal of "selective execution of certain parts, or all of a program. . . ." *Id.* at 3:28–40.

1. The Specification Makes Clear that "Fake" Emulation Is Not Required by the '115/'322 Patents

As discussed at pages 27–29 of Columbia's opening brief, emulation in the '115/'322 patent does not require simulation. Symantec claims that "[t]he core of emulation [is] that the system running the emulated software is not real, but is rather … fake[.]" Symantec Brief at 26. But that is not the core of the '115/'322 emulator. The '115/'322 patents describe emulators that can analyze the actual running code in the executable, not "fake" code. For example, an "instruction-level emulator" is used that is "compiled in the code," which means that the

emulator simply becomes part of the executable. *Id.* at 14:6–9; Szajda Decl. II, ¶ 22. The executable is not "fake."

As a further example, the emulator can act in the same way as "a modern debugger." In fact, it can use the debugging facilities that already exist on the system. *Id.* at 13:9–15. Debuggers do not act on "fakes." Debuggers analyze actual code. Ex. HH ("The purpose of a debugger such as GDB is to allow you to see what is going on 'inside' another program while it executes—or what another program was doing at the moment it crashed."); Szajda Decl. II, ¶ 23.

The specification consistently discusses embodiments in which portions of the actual code is monitored and executed by the emulator. In these sections there is no reference to the code being "fake" or "not real." For example, some embodiments act by "monitoring the one or more applications," not fake versions of the applications. Ex. G, '115 patent at 12: 27–45; Fig. 3, Blocks 310–320; *see also id.* at 16:47–67; Fig. 6, Blocks 610–620.

1. Virtualization Is an Alternative Feature

After an anomaly is detected, the patents describe a strategy for repairing any damage caused by that anomaly. This process is called error virtualization. For example, the specification states that "[o]nce an anomaly is detected . . . error virtualization may be used to reverse (undo) the effects. . . ." *Id.* at 3:57–60. The specification repeatedly states that virtualization occurs after detection of the anomaly. *See, e.g.*, *id.* at 10:36–41. Indeed, the portions of the specification from 13:61 through 14:5 referenced by Symantec relate to step 360 in Figure 3, which occurs ***after*** detection of an anomaly in step 320. The surrounding passages make clear that the virtualization is part of the repair process, not the detection process.[12] *Id.* at

[12] Symantec also points to a reference to "virtual register" at 14:51–56. Symantec Brief at 24. But this passage occurs in a paragraph that is on its face about one of multiple possible implementations: "[f]or example, the instruction-level emulator ***may*** be implemented . . . ." Ex. G, '115 patent at 14:45–48 (emphasis added).

15:33–56.

The portion of Appendix A to the provisional application cited by Symantec confirms that simulation and error virtualization are features that *can* be present, not features that *must* be present. Symantec Brief at 24, 26. Appendix A, which is a paper, makes clear that detection of anomalies and error virtualization are distinct concepts: "Our description of the systems raises several questions . . . . Can the system detect real attacks and faults and react to them? How effective is our 'error virtualization'. . .?" Ex. 12, at COL00007631.

2. Symantec's Extrinsic Evidence Ignores the Unique Features of the '115/'322 Patents

Symantec confirms that the specification provides a "clear disclosure" of emulator, making the use of extrinsic evidence to alter that meaning inappropriate. Symantec Brief at 25. Nonetheless, Symantec presents extrinsic evidence that it claims supports its construction. Symantec's extrinsic evidence does not take account of the unique role the emulator plays in the '115/'322 patents. Moreover, many dictionaries are inconsistent with Symantec's position and confirm that simulation and emulation are different.

One of the insights of the patents is the ability to very efficiently monitor and execute only selective portions of a program's code. *See, e.g.*, Ex. G, '115 patent at 13:14–24; 9:12–15; 16:47–17:24; Fig. 6. These passages from the specification make clear that a classical emulation system in which an entire program is analyzed in a separate environment is not necessary.

Symantec's references do not account for the role played by the emulator in the '115/'322 patents. *Phillips*, 415 F.3d at 1321 ("The main problem with elevating the dictionary to such prominence is that it focuses the inquiry on the abstract meaning of words rather than on the meaning of claim terms within the context of the patent.").

For example, Symantec relies on a patent that provides its own formal definition of

emulator. Ex. 14, at 2:41–45. But if the term emulator had a generally accepted meaning as Symantec suggests, no special definition would have been necessary. Moreover, Symantec's Ex. 14 is clearly not describing the unique selective execution taught in the '115/'322 patents. *Id.* at 13:14-25 (teaching the emulation of isolated portions or slices of code). The same holds for Symantec's other technical references, such as Exhibits 17 and 20, which are directed at executing an entire program. Ex. 17, at 453 ("a program [is] executed."); Ex. 20, at 46 (discussing analyzing an entire program). Indeed, many extrinsic sources make clear that simulator and emulator have significantly different meanings. *See, e.g.*, Exs. I, FF. And one of Symantec's dictionaries confirms that emulation should be "constrast[ed] with simulation." Ex. GG at 157.

B. "anomaly"

Symantec's attempt to import the limitation a model of "typical, attack free computer system usage" into the claims contradicts the specification and violates the express instructions of the Federal Circuit as discussed at pages 20–23 of Columbia's opening brief.

Symantec advances a false syllogism: anomaly detectors detect departures from normal activity and, therefore, according to Symantec, the information used to build the models used to detect anomalies must be 100% normal information without any supplemented abnormal information. The claims of the '115/'322 patents, however, confirm that this argument cannot be correct. The '115 patent independent claim 11 recites a "method of detecting anomalous program executions," while dependent claim 18 recites "where the model ***reflects attacks*** against at least part of the program." Ex. G, '115 patent at claim 18. The same pattern exists for '115 patent independent claim 32 and dependent claim 39. What these claims establish is that detection of departures from normal activity can consider abnormal information. Indeed, any

other conclusion would render the dependent claims 32 and 39 nonsensical.[13] *Alcon Research, Ltd. v. Apotex Inc.*, 687 F.3d 1362, 1367 (Fed. Cir. 2012) ("It is axiomatic that a dependent claim cannot be broader than the claim from which it depends."). There is no support for Symantec's attempt to import the restriction to "typical, attack free" data into the claims.

Columbia anticipates that Symantec will argue that Columbia's construction is not sufficient because a jury will not know what standards are used to determine whether an anomaly has occurred. Symantec Brief at 20. This is incorrect. An anomaly is behavior that deviates from normal and may correspond to an attack. The patent claims make clear that the tool used to detect this is a "model of function calls for at least a part of the program." Ex. G, '115 patent at claims 1, 11, 21, 22, 32, and 42. The claims do not require a model of normal behavior that ***only*** includes data on normal activity.

C. "application community"

Symantec wants to import into the definition of application community the concept that the members of the community must all be running the "***modeled*** program." This limitation, however, is not appropriate because it ignores the different roles members of an application community can play. For example, members of an application community can share models, but in the alternative, they can also provide information that is used to update unique models maintained by each member: "In particular, some embodiments can share models with each other and/***or update each other's models*** such that the learning of anomaly detection models is relatively quick." Ex. G, '115 patent at 6:33–37 (emphasis added). The specification also consistently discusses embodiments in which only a portion of a program is run and studied at each member of the community: "For example, for each member of an application community,

[13] The fact that the issued claims refer to models of "functions" does not allow for the importation of additional limitations. Symantec Brief, at 22. The patent discloses modeling function calls without any reference to only normal activity. Ex. G, '115 patent at 1:65–2:4.

some particular randomly chosen function or functions and its associated data may be chosen for modeling, while others may simply be ignored." *Id.* at 7:4–8; *see also id.* at Fig, 6; 16:54–67; 17:1–23. Indeed, the claims and specification contemplate that different models may exist that are combined together. *See, e.g., id.* at 8:9–31; claim 9. The claims also make clear that the model may be of only part of a program. *See, e.g., id.* at claim 8. Columbia is concerned that Symantec's phrase "modeled program" does not take into account the diversity of roles played by the application community. Specifically, Symantec's construction appears to require that each member of the community run the exact same model, which is obviously inconsistent with the specification. Symantec's construction also appears to ignore that there must be additional infrastructure to support an application community besides simply running the modeled program. The application community also consists of structures that manage the generation and deployment of machine learning models. *See id.* at 16:55–58 ("Each portion or slice of the application's code may, for example, be assigned to one of the members of the application community (e.g., ***workstation, server, etc.***)."") (emphasis added); 17:54–63 (describing a "controller" member of the application community).

Columbia's construction, when read in light of the claims, requires that the members of the application community either (a) run the same modeled application or a portion thereof; (b) run an application that allows them to share information that is used to build a model. Columbia believes this is clear, but if it is not, Columbia is not opposed to additional clarification being added to its construction. However, Columbia cannot agree to the insertion of the term "modeled" to the extent this excludes many embodiments in the specification.

Dated: August 28, 2014

Respectfully submitted,

By: */s/ Dana D. McDaniel*
Dana D. McDaniel (VSB No. 25419)

dmcdaniel@spottsfain.com
John M. Erbach (VSB No. 76695)
jerbach@spottsfain.com
Spotts Fain, P.C.
411 East Franklin Street, Suite 600
Richmond, Virginia 23219
Phone: (804) 697-2065
Fax: (804) 697-2165

IRELL & MANELLA LLP
David I. Gindler (dgindler@irell.com)
Jason G. Sheasby (jsheasby@irell.com)
Richard M. Birnholz (rbirnholz@irell.com)
1800 Avenue of the Stars
Suite 900
Los Angeles, California 90067-4276
Phone: (310) 277-1010
Fax: (310) 203-7199
Pro Hac Vice

ATTORNEYS FOR PLAINTIFF
THE TRUSTEES OF COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

CERTIFICATE OF SERVICE

I hereby certify that on the 28th day of August, 2014, I will electronically file the foregoing with the Clerk of Court using the CM/ECF system, which will then send a notification of such filing (NEF) to all counsel of record:

/s/

Dana D. McDaniel (VSB No. 25419)
dmcdaniel@spottsfain.com
Spotts Fain, P.C.
411 East Franklin Street, Suite 600
Richmond, Virginia 23219
Phone: (804) 697-2044
Fax: (804) 697-2144